IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 November 2022, accepted 21 December 2022, date of publication 23 December 2022, date of current version 29 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3231889

RESEARCH ARTICLE

Portfolio Management Framework for Autonomous Stock Selection and Allocation

JAE-SEUNG KIM, SANG-HO KIM, AND KI-HOON LEE
School of Computer and Information Engineering, Kwangwoon University, Nowon-gu, Seoul 01897, Republic of Korea

Corresponding author: Ki-Hoon Lee (kihoonlee@kw.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant NRF-2022R1F1A1062787; and in part by the Research Grant of Kwangwoon University, in 2021.

ABSTRACT Portfolio management is essential to reduce risks and maximize profits. It can be classified into two processes: stock selection and allocation. Stock selection identifies stocks with high expected profits, whereas stock allocation determines the investment ratios for the selected stocks. Most current stock selection methods employ a ranking approach that predicts a ranked stock list based on the relations between stocks. However, the ranking-based stock selection methods do not consider the stock allocation problem. Furthermore, the methods use either simple graphs or hypergraphs, but not both. The sole use of simple graphs or hypergraphs induces information loss as the collective or pairwise relations, respectively, are disregarded. To overcome these issues, we propose a novel portfolio management framework called ASA that combines ranking models with classification and regression models for autonomous stock selection and allocation. For stock selection, the simple graph- and hypergraph-based ranking models are hybridized for relational modeling to select the most profitable stocks. For stock allocation, the classification and regression models are combined to determine the investment ratio. Furthermore, ASA extracts robust features using hierarchical clustering, feature selection, and dimensionality reduction, following which it captures temporal information using long short-term memory (LSTM), bidirectional LSTM, and the Hawkes attention mechanism. The performance of ASA is compared with that of deep learning-based state-of-the-art methods. The experimental results for stocks included in the Standard & Poor's 500 index demonstrate that ASA achieves a compounded annual growth rate of 58.2%, which is 39.1%P higher than that of the second-best performing method.

INDEX TERMS Portfolio management, stock selection, stock allocation, deep learning, graph neural network, simple graph, hypergraph.

I. INTRODUCTION

A stock portfolio is a blended combination of various stocks [1], and investors can reduce risks and maximize profits through effective portfolio management. Portfolio management can be achieved using two types of processes: (1) stock selection and (2) stock allocation [2]. Stock selection is the process of selecting stocks that comprise a portfolio, whereas stock allocation is the process of determining the proportion of investment in the selected stocks.

The associate editor coordinating the review of this manuscript and approving it for publication was Bing Li.

In the field of investments, deep learning-based methods have exhibited superior performance over statistical and traditional machine learning methods [3], [4], [5]. Portfolio management using deep learning is typically formulated as a classification or regression problem for stock selection. Classification- or regression-based methods [6], [7], [8], [9], [10], [11], [12] predict the future trends or prices of individual stocks and select the top-ranked ones that exhibit the highest possibility of an uptrend or the highest return rate. Most current stock selection methods [13], [14], [15], [16], [17], [18], [19] have adopted a ranking approach that predicts a ranked stock list, where stocks with higher rankings are expected to achieve higher return rates. Ranking-based methods have yielded promising results compared to classification- and regression-based methods by directly optimizing the ranking of profitable stocks.

In contrast to the stock selection problem, deep learning-based stock allocation has not been sufficiently researched. In ranking-based stock selection methods, top-ranked stocks are invested in at the same proportion, which is not optimal. For example, Table 1 compares the return rates for different investment ratios (or weights). If the top four stocks are invested in equally, the return rate is 2%. However, if near-optimal weights can be predicted, a 2%P (percent point) higher return rate can be obtained.

TABLE 1. Different stock allocations and their return rates.

Stocks	S_1	S_2	S_3	S_4	Return rates
Actual return rates	+5.5%	+2.5%	−2.0%	+2.0%	
Predicted ranking	1	2	3	4	
Equal weights	0.25	0.25	0.25	0.25	+2.0%
Predicted weights	0.60	0.20	0.05	0.15	+4.0%

The majority of current ranking-based stock selection methods [13], [14], [15], [16], [17], [18], [19] have used either simple graphs or hypergraphs to represent the pairwise or collective relations among stocks, respectively. However, the sole use of simple graphs or hypergraphs incurs information loss because one type of relation is disregarded. For example, Fig. 1 depicts second-order relations whereby Oracle (ORCL), Tesla (TSLA), and Apple (AAPL) are related via Larry Ellison who founded Oracle in 1977, and became a board member of Tesla in 2018 and of Apple in 1997. In a simple graph, two different pairwise relation instances are created by the relation types "Founded By" and "Board Member": (1) ORCL and TSLA and (2) ORCL and AAPL, resulting in a loss of collective relations. In a hypergraph, the two pairwise relation instances are represented as a single relation instance (ORCL, TSLA, and AAPL), resulting in a loss of pairwise relations.

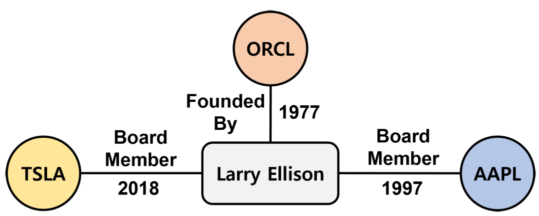

FIGURE 1. An example of the second-order relation.

In this study, we propose a novel portfolio management framework called ASA for autonomous stock selection and allocation that addresses the aforementioned shortcomings of ranking-based methods. Fig. 2 summarizes the framework. First, we combine the ranking models for stock selection with the classification and regression models for stock allocation. Our key finding is that the ranking, classification, and regression models can successfully complement one another for effective portfolio management. Second, we hybridize

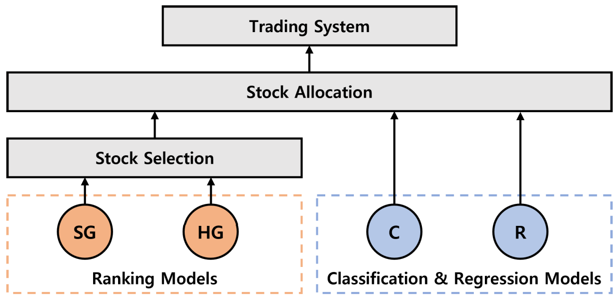

FIGURE 2. Our framework.

simple graph- and hypergraph-based ranking models for relational modeling in stock selection. Third, we extract robust features using hierarchical clustering, feature selection, and dimensionality reduction. Fourth, we extract the temporal information from time-series features using long short-term memory (LSTM) [20], bidirectional LSTM (Bi-LSTM) [21], and the Hawkes attention mechanism [14]. Fifth, compared with deep learning-based state-of-the-art methods (HATS [6], STH [14], and RSR [13]), ASA achieves a compounded annual growth rate (CAGR) of 58.2%, which is 39.1%P higher than that of the second-best performing method for stocks that are included in the Standard & Poor's 500 (S&P500) index, which consists of the 500 largest companies in the United States.

The remainder of this paper is organized as follows. Section II introduces graph neural networks (GNNs), and Section III reviews related work. Sections IV and V present ASA and the experimental results, respectively. Finally, the conclusion and suggestions for future work are outlined in Section VI. For ease of reading, Table 8 in the Appendix lists the abbreviations used in this study.

II. BACKGROUND

A. TERMINOLOGIES

A simple graph G_s is defined as $G_s = (V, E)$, where V is the set of nodes and E is the set of edges. A square adjacency matrix is used to represent the pairwise relations between nodes in G_s. A hypergraph G_h is a generalization of the graph that can represent the collective relations among nodes. Mathematically, it is defined as $G_h = (V, E)$, where V is the set of nodes and E is the set of hyperedges. Each hyperedge defines a relation instance among an entire set of nodes (i.e., a collective group). An $N \times M$ incidence matrix H is used to link the nodes to their hyperedges [22], where N is the number of nodes and M is the number of hyperedges. The value of $H(p, q)$ is 1 if a node p is in a hyperedge q (i.e., $p \in q$) and 0 otherwise. Two diagonal degree matrices can be defined from the incidence matrix [22]. First, the $N \times N$ node degree matrix D_v counts the number of hyperedges to which each node belongs. Second, the $M \times M$ hyperedge degree matrix D_e counts the number of nodes in each hyperedge.

B. GNN

GNNs [23] learn the node representations using the relations between nodes in the simple graph (or hypergraph). The operational procedure of a GNN can be divided into two processes [24]: (1) *AGGREGATE* and (2) *UPDATE*. The *AGGREGATE* function, which is learnable, iteratively aggregates information from the neighboring nodes of each node p and generates the representation $h^i_{N(p)}$ for the set $N(p)$ of neighboring nodes, as indicated in Equation (1). The *UPDATE* function, which is also learnable, generates the representation h^i_p of node p by combining h^{i-1}_p at iteration $i-1$ with $h^i_{N(p)}$, as expressed by Equation (2).

$$h^i_{N(p)} = AGGREGATE(\{h^{i-1}_q : q \in N(p)\}) \tag{1}$$

$$h^i_p = UPDATE(h^{i-1}_p, h^i_{N(p)}) \tag{2}$$

With the development of GNNs, advanced networks have been proposed [25], such as the graph convolutional network (GCN) [26] and graph attention network (GAT) [27]. The GCN uses the convolutional neural network to capture the local connection patterns, whereas the GAT uses the self-attention mechanism [28] to assign different weights based on the importance of the neighboring nodes.

C. GAT

To capture the relative importance of the neighboring nodes of node p (and p itself), GAT assigns different weights to them using the attention mechanism. In the GAT, the *AGGREGATE* and *UPDATE* functions in the GNN are not separated. The GAT calculates an attention coefficient α_{pq} between nodes p and q, as indicated by Equation (3), where h^{i-1}_p and h^{i-1}_q denote the representation of nodes p and q at iteration $i-1$. Moreover, a and W are learnable parameters, ϕ_1 is a nonlinear function such as LeakyReLU, and $\|$ is the concatenation operation. The attention coefficient α_{pq} indicates the importance of q to p in the simple graph [29]. The hypergraph attention network is described in Section IV-G.

$$\alpha_{pq} = \frac{\exp(\phi_1(a\left[Wh^{i-1}_p \| Wh^{i-1}_q\right]))}{\sum_{k \in N(p) \cup p} \exp(\phi_1(a\left[Wh^{i-1}_p \| Wh^{i-1}_k\right]))} \tag{3}$$

Based on the attention coefficients, the GAT obtains a new representation h^i_p of node p as a weighted sum of the neighboring nodes $N(p)$ including itself, as indicated in Equation (4), where α_{pq} is the attention coefficient that is defined in Equation (3) and ϕ_2 is a nonlinear function such as sigmoid or softmax.

$$h^i_p = \phi_2(\sum_{q \in N(p) \cup p} \alpha_{pq} W h^{i-1}_q) \tag{4}$$

To stabilize the learning process, the GAT employs a multi-head attention mechanism, as expressed by Equation (5), where K is the number of heads.

$$h^i_p = \phi_2(\frac{1}{K}\sum_{k=1}^{K} \sum_{q \in N(p) \cup p} \alpha^k_{pq} W^k h^{i-1}_q) \tag{5}$$

III. RELATED WORK

Our study is related to recent work on classification, regression, ranking, and reinforcement learning methods in the investment field.

A. CLASSIFICATION AND REGRESSION METHODS

Classification-based methods [6], [7], [8], [9], [10] that predict the price movements of individual stocks have been proposed for stock selection. Kim et al. [6] proposed a hierarchical graph attention network called HATS to aggregate information from the neighboring nodes and different relation types effectively. Sawhney et al. [7] presented a spatio-temporal hypergraph convolution network to learn the related stock movements. Yuan et al. [8] integrated two random forest algorithms to select features and predict stock price trends. Ye et al. [9] combined a GCN and gated recurrent unit (GRU) [30] to learn multiple relations between stocks. Chen and Robert [10] proposed a multi-graph recurrent network to learn from both text and relational data.

Regression-based methods [11], [12] that predict the future values of individual stocks have been developed for stock selection. Sun [11] predicted the earnings yield of companies using sequence prediction models. Yang [12] combined the predicted return rate with fundamental factors such as the profitability, leverage, and liquidity.

B. RANKING METHODS

Ranking-based methods [13], [14], [15], [16], [17], [18], [19] that predict a ranked stock list have been proposed for stock selection. Feng et al. [13] presented a relational stock ranking framework called RSR that uses a temporal graph convolution to consider the relations between stocks in a time-sensitive manner. Sawhney et al. [14] proposed a spatio-temporal hypergraph attention network called STHAN-SR, which combines spatial hypergraph convolutions with the Hawkes attention mechanism to capture spatial and temporal dependencies. Gao et al. [15] proposed a time-aware relational attention network to capture the time-varying strength of relations between stocks. Feng et al. [16] presented a new GAT, the attention mechanism of which is based on a GCN, to capture the local correlation topology information. He et al. [17] proposed a static-dynamic GNN to capture the potential relations between stocks. Heyuan et al. [18] proposed an adaptive long-short pattern transformer to capture time factors at different context scales. Ma et al. [19] developed an attribute-driven fuzzy hypergraph network to describe the strength of the collective relations and to simulate the influence of stocks.

Although the aforementioned studies have achieved considerable performance improvements in stock selection, they did not consider the problem of stock allocation.

C. REINFORCEMENT LEARNING METHODS

Reinforcement learning methods [31], [32], [33], [34], [35] that determine the proportion of investment or trading actions in predetermined stocks have been proposed. Ye et al. [31] proposed a state augmented reinforcement learning framework to combine different data types, such as news and stock prices. Koratamaddi et al. [32] presented a sentiment-aware deep reinforcement learning framework to consider the overall attitude of investors. Park and Lee [33] incorporated imitative reinforcement learning, multi-task learning, multistep learning, and dynamic delay for algorithmic trading. Kim, Park, and Lee [34] hybridized two deep reinforcement learning algorithms to determine the trading actions and stop-loss boundaries. Lim, Cao, and Quek [35] proposed a reinforcement learning agent that uses LSTM to reduce the time lag of technical indicators by predicting the future stock prices. However, these studies did not consider the stock selection problem.

IV. PORTFOLIO MANAGEMENT FRAMEWORK

We propose a novel portfolio management framework called ASA for autonomous stock selection and allocation.

A. OVERVIEW OF FRAMEWORK

The proposed framework for portfolio management is depicted in Fig. 2. It autonomously selects stocks and allocates the selected stocks by combining ranking, classification, and regression models. For the stock selection, a simple graph-based ranking model *SG* is combined with a hypergraph-based ranking model *HG*. This is because the simple graph, which cannot represent the collective relations among stocks, complements the hypergraph, which ignores the pairwise relations between stocks. For the stock allocation, the classification and regression models are combined to determine the investment ratio. This is because the classification model, which predicts the future trend of each stock, complements the regression model, which predicts the future price of each stock.

B. ALGORITHM

Algorithm 1 Stock Selection and Allocation Algorithm

Input: (1) the ranking models *SG* and *HG*,
(2) the classification model C,
(3) the regression model R

Output: (1) the selected stock vector $S_{selected}$,
(2) the investment ratio vector W_{ratio}

1. Initialize weight vectors W_C and W_R.
2. $S_{selected} \leftarrow topK(SG) \cap topK(HG)$
3. **for** $s_i \in S_{selected}$ **do**
4. Append the predicted uptrend probability $C(s_i)$ to W_C.
5. Append the predicted return rate $R(s_i)$ to W_R.
6. **end for**
7. Normalize the weight vectors W_C and W_R.
8. Calculate the investment ratio vector: $W_{ratio} \leftarrow COMBINE(W_C, W_R)$.
9. Normalize the investment ratio vector W_{ratio}.
10. **return** $S_{selected}, W_{ratio}$

Algorithm 1 presents the stock selection and allocation algorithm for portfolio management. We intersect the top-K stocks that are obtained from the ranking models *SG* and *HG* to select stocks that are strongly expected to generate profits (line 2). For the stock allocation, we predict the uptrend probability and return rate of each selected stock using the classification and regression models C and R, and store each of them in the weight vectors W_C and W_R (lines 3 to 6). We normalize the weight vectors using the softmax function (line 7). We apply a *COMBINE* function that integrates the weight vectors to calculate the investment ratio vector W_{ratio} (line 8). We employ the element-wise multiplication ($\odot$) as the *COMBINE* function. Finally, we normalize W_{ratio} to make the sum of the investment ratios be 1 and return the selected stocks and their investment ratios (lines 9 to 10).

The ranking models as well as classification and regression models that are used in this algorithm can be extended to the case of multiple models by applying an ensemble approach. Various functions other than the intersection are possible in line 2, and various *COMBINE* functions other than the element-wise multiplication are possible in line 8. We leave those topics for future studies because the focus of this study is the portfolio management framework, which can be extended to accommodate various models and functions.

C. ARCHITECTURE

Fig. 3 depicts the detailed architecture of our framework. The proposed architecture preprocesses stock and economic data, applies temporal modeling to capture dependencies in the historical data, selects stocks using ranking models with relational modeling, and allocates the selected stocks using the classification and regression models. We employ a shallow network for the temporal modeling that is connected to the relational modeling because an excessive depth may lead to unstable learning and overfitting. Each component is explained in detail in the following subsections.

D. DATA PREPROCESSING

We use the candlestick components and technical indicators of each stock as well as various economic data such as commodities, bonds, currencies, and market indices in data preprocessing. Fig. 4 presents the data preprocessing procedure. We calculate the candlestick components and technical indicators listed in Table 2 using stock data that consist of opening, high, low, closing, and volume values. Thereafter, we normalize the input features using a robust standardization method [36], which minimizes the effect of outliers using the median value and interquartile range. Subsequently, we cluster all features using an agglomerative hierarchical clustering method to classify features that are similar in terms of the Pearson's correlation coefficient. Following clustering, we select features for each cluster if the number of features in the cluster is greater than the threshold F, as illustrated

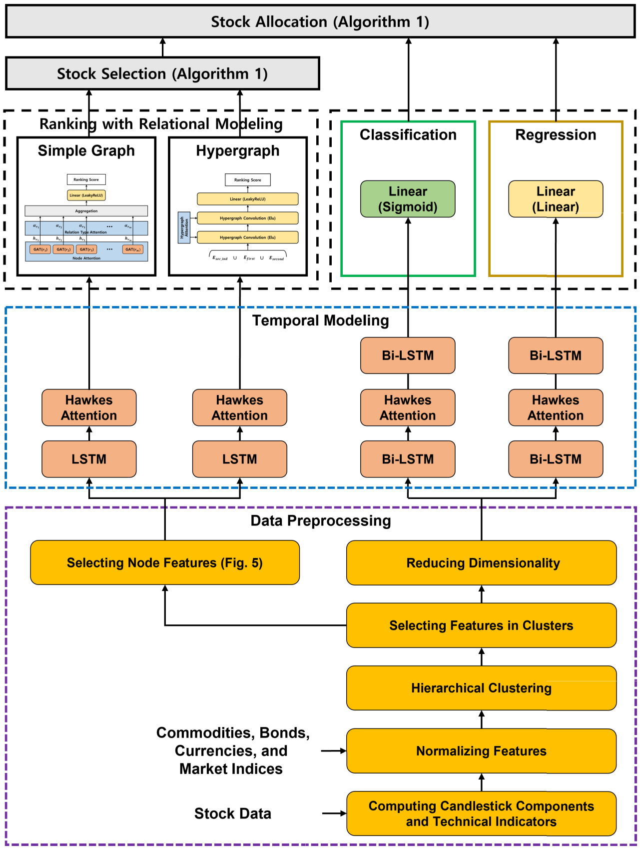

FIGURE 3. Architecture of ASA.

in Fig. 4, using a sequential forward feature selector [37]. The selected features for the clusters are merged, and the merged features may differ for each stock as well as the classification and regression models of each stock, as indicated in Table 3. After merging the selected features, the dimensionality of the merged features is reduced to remove noise using sparse principal component analysis [38], and the final output is used in the temporal modeling for classification and regression.

For relational modeling, it is necessary to provide an identical set of features for each node of a simple graph and hypergraph. To this end, we propose the node feature selection method illustrated in Fig. 5. To obtain the common features for each stock, we obtain the merged features F_i^C and F_i^R for the classification and regression models C_i and R_i of stock i, respectively. We obtain the common features by intersecting F_i^C and F_i^R for each stock. Subsequently, we combine the common features of all stocks and select the top-K features with the highest frequencies to obtain an identical set of features. Table 3 presents an example of the node feature selection process. The common features that intersect the merged features, F_i^C and F_i^R, for each stock are (open and high), (open), and (open, high, and AD), respectively. We combine the common features of all stocks and select the top two features (i.e., open and high) with the highest frequencies.

TABLE 2. Input features.

Feature group	Features
Candlestick components [39]	the lengths of upper shadow line, lower shadow line, and body; the body color
Overlap studies [40]	BBANDS, DEMA, EMA, HT-TRENDLINE, KAMA, MA, MAMA, MIDPOINT, MIDPRICE, SAR, SAREXT, SMA, T3, TEMA, TRIMA, WMA
Momentum indicators [40]	ADX, ADXR, APO, AROON, AROONOSC, BOP, CCI, CMO, DX, MACD, MACDEXT, MACDFIX, MFI, MINUS_DI, MINUS_DM, MOM, PLUS_DI, PLUS_DM, PPO, ROC, ROCP, ROCR, RSI, STOCH, STOCHF, STOCHRSI, TRIX, ULTOSC, WILLR
Volume indicators [40]	AD, ADOSC, OBV
Volatility indicators [40]	ATR, NATR, TRANGE
Commodities	Gold, Silver, Copper, Crude Oil
Bonds	Treasury Yield (13 Weeks, 2 Years, 5 Years, 10 Years, 30 Years)
Currencies	KWR/USD, EUR/USD, GBP/USD, JPY/USD, CNY/USD, BTC/USD
Market indices	S&P500, Dow Jones, Nasdaq, Nikkei225

TABLE 3. An example of node feature selection.

Stocks	Models	Merged features	Common features	Top-2 features
S_1	C_1	open, high, low, MA	open, high	open, high
	R_1	open, high, close, T3		
S_2	C_2	open, low, ADX, MA	open	
	R_2	open, close, SAR, TRIX		
S_3	C_3	open, high, AD, CCI	open, high, AD	
	R_3	open, high, AD, TEMA		

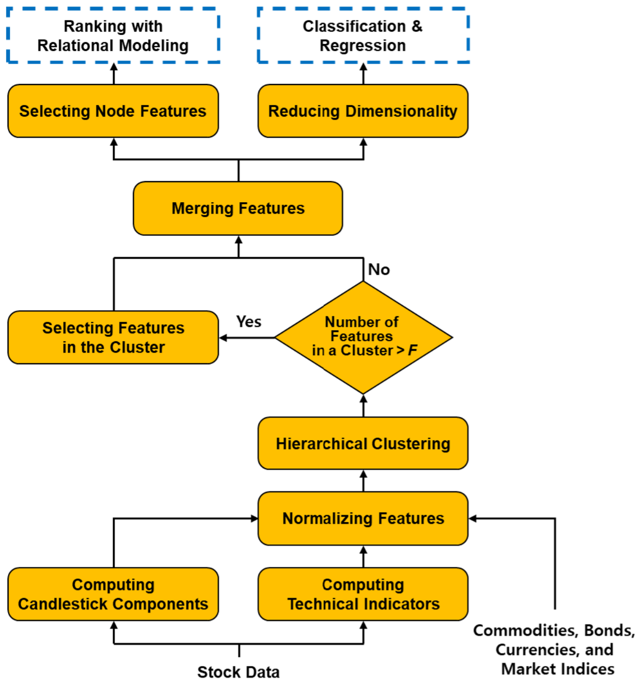

FIGURE 4. Data preprocessing.

E. TEMPORAL MODELING

The input for the temporal modeling is a sliding window of the preprocessed feature vectors. The temporal modeling

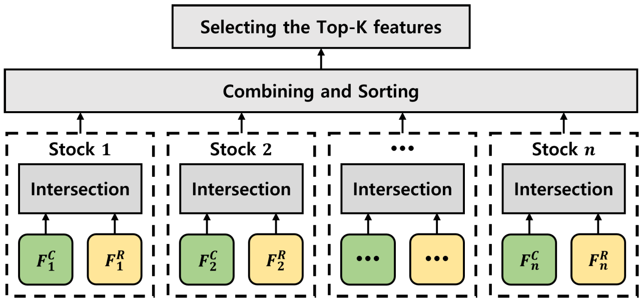

FIGURE 5. Method for node feature selection.

process differs for the stock selection, which uses ranking with relational modeling, and stock allocation, which uses classification and regression, as depicted in Fig. 3. For the stock selection, an LSTM layer and Hawkes attention mechanism are applied, and the output of the Hawkes attention for each stock is used as the node feature vector for the ranking models. For the stock allocation, the Hawkes attention mechanism is applied between the first and second Bi-LSTM layers, and the output of the second Bi-LSTM layer is used for the classification and regression models.

The Hawkes process is a self-exciting temporal point process that models the time-decaying influence of events on the future [41]. The Hawkes attention mechanism combines the attention mechanism with the Hawkes process to assign higher weights to important days that influence future prices as well as to capture the self-exciting phenomenon in the stock and economic data, as depicted in Fig. 6.

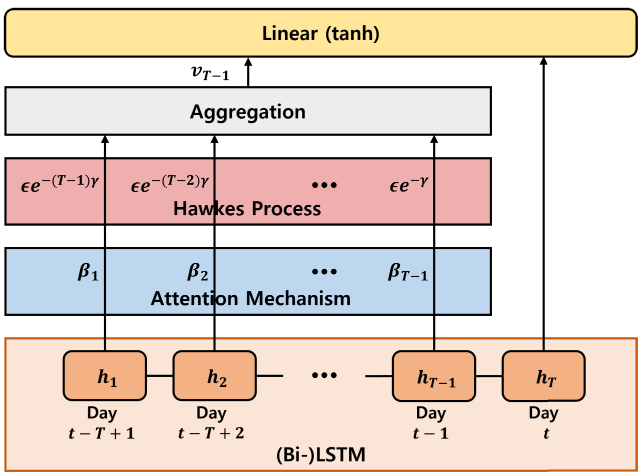

FIGURE 6. Hawkes attention mechanism.

The Hawkes attention mechanism first calculates the day-level latent representation λ_τ for each time step $\tau < T$, as expressed by Equation (6), where β_τ is the learnable attention weight, h_τ is the hidden state, and h_T is the last hidden state. Subsequently, it calculates the vector v_{T-1} by applying the Hawkes process to λ_τ of the previous $T-1$ time steps, as indicated in Equation (7), where ϵ and γ are the learnable excitation and decay parameters, respectively, and Δt_τ is the time interval between the current T and past time step τ. After applying the Hawkes process, we combine h_T with v_{T-1} and then obtain the temporal feature vector h_t^{haw} at day t via a linear layer with the tanh activation function.

$$\lambda_\tau = \beta_\tau h_\tau, \ \beta_\tau = \frac{\exp(h_\tau W h_T)}{\sum_{\tau=1}^{T-1} \exp(h_\tau W h_T)} \tag{6}$$

$$v_{T-1} = \sum_{\tau=1}^{T-1} (\lambda_\tau + \epsilon \max(\lambda_\tau, 0) e^{-\gamma \Delta t_\tau}) \tag{7}$$

$$h_t^{\text{haw}} = \tanh(W[h_T \| v_{T-1}] + b) \tag{8}$$

F. SIMPLE GRAPH-BASED RANKING MODEL

We use three types of relations: the (1) sector, (2) first-order, and (3) second-order corporate relation types to construct the simple graph. A sector is a collection of industries that have similar characteristics [42]. For example, the IT sector consists of industries such as IT services, software, and communications equipment. We collect the sector-industry and corporate relations from the global industry classification standard [43] and Wikidata [44]. A sector relation type represents the relation instances between two stocks in the same sector. A first-order relation type R_1 represents the relation instances between two stocks that are connected via R_1. A second-order relation type R_2R_3 represents the indirect relation instances through an entity E that bridges two stocks A and B via entity-relations R_2 and R_3 (i.e., $A \overset{R_2}{-} E \overset{R_3}{-} B$). We create each relation instance as a bidirectional edge in the graph. The relation types that are used in this study, which include those used in [6] and [13], are comprehensively listed in Table 7 in the Appendix. Graphs can be constructed using other data sources and methods, such as news [9] and dynamic time warping [18], which we leave for future studies.

Fig. 7 presents the architecture of the simple graph-based ranking model. We apply the attention mechanism hierarchically at the node and relation type levels to capture the relative importance of the neighboring nodes as well as that of the

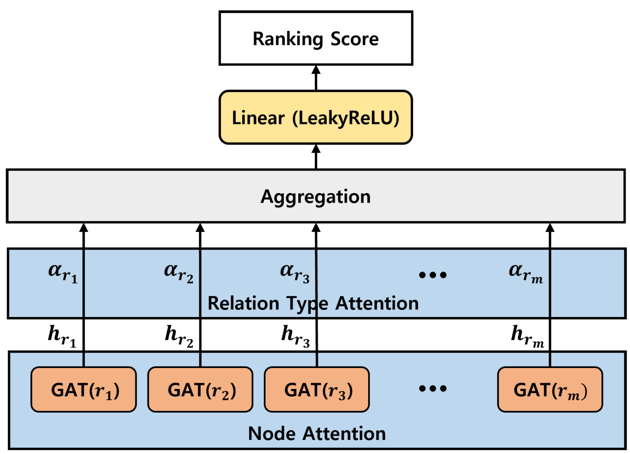

FIGURE 7. Architecture of simple graph-based ranking model.

different relation types. At the node level, we use a GAT to assign weights to the neighboring nodes for each relation type r_i and obtain the relational feature vector h_{r_i}. At the relation type level, we calculate the attention coefficient α_{r_i}, as expressed by Equation (9), where W and b are learnable parameters and m is the number of relation types. After calculating the attention coefficients, we obtain the final relational feature vector v_r by aggregating the weighted h_{r_i}, as indicated in Equation (10). Subsequently, we predict the ranking score via a linear layer with the LeakyReLU activation function.

$$\alpha_{r_i} = \frac{\exp(h_{r_i} W + b)}{\sum_{k=1}^{m} \exp(h_{r_k} W + b)} \tag{9}$$

$$v_r = \frac{1}{m} \sum_{k=1}^{m} \alpha_{r_k} h_{r_k} \tag{10}$$

We employ the loss function [13] in Equation (11), where $\hat{r}$ and r are the predicted and ground-truth ranking vectors, respectively, to optimize the simple graph-based ranking model. The ground-truth ranking score is calculated using the one-day return rate of each stock. The loss function combines (1) pointwise regression loss and (2) pairwise ranking-aware loss. In Equation (11), ρ is a hyperparameter that balances the two loss terms, and N is the number of stocks.

$$L = \|\hat{r} - r\|^2 + \rho \sum_{i=0}^{N} \sum_{j=0}^{N} \max(0, -(\hat{r}_i - \hat{r}_j)(r_i - r_j)) \tag{11}$$

G. HYPERGRAPH-BASED RANKING MODEL

We convert the sector, industry, first-order, and second-order relation instances into hyperedges to construct a hypergraph. First, for the sector and industry relation instances, a set of stocks belonging to the same sector or industry is converted into a hyperedge $e \in E_{sec\_ind}$. Second, for the first-order relation type R_1 on a stock A, the set of stocks that are connected to A via R_1 is converted into a hyperedge $e \in E_{first}$. Third, the set of stocks that are connected to an entity E via relation types R_2 and R_3 is converted into a hyperedge $e \in E_{second}$. These hyperedges are combined to form $E = E_{sec\_ind} \cup E_{first} \cup E_{second}$ with duplication elimination.

Fig. 8 depicts the architecture of the hypergraph-based ranking model. To extract the relational features from the hypergraph, we employ the hypergraph convolution (HConv) with a multi-head attention mechanism [45], as expressed by Equation (12), where X is the input feature matrix, H_{attn}^k is the hypergraph incidence matrix with attention edge weights, P^k is a learnable parameter matrix, K is the number of heads, and D_v and D_e are the node and hyperedge degree matrices, respectively. We employ two HConv layers with an ELU activation function and a final linear layer with a LeakyReLU activation function. At the first HConv layer, X is the temporal feature matrix that is obtained from the temporal modeling. The hypergraph-based ranking model is optimized using the

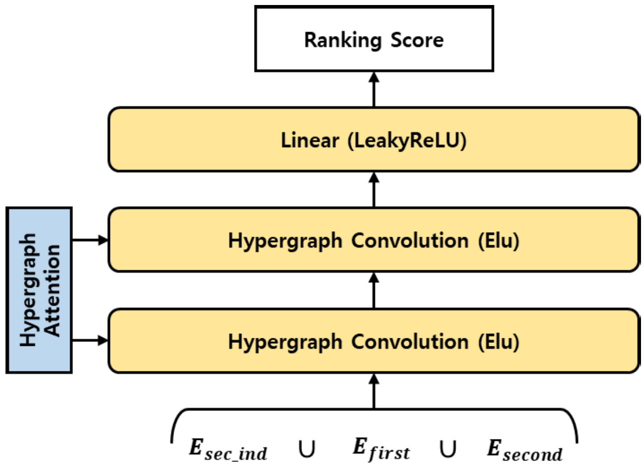

FIGURE 8. Architecture of hypergraph-based ranking model.

loss function defined by Equation (11).

$$\begin{aligned} X' &= \frac{1}{K} \sum_{k=1}^{K} \text{HConv}(X, H_{\text{attn}}^k, P^k) \\ &= \frac{1}{K} \sum_{k=1}^{K} \text{ELU}(D_v^{-\frac{1}{2}} H_{\text{attn}}^k D_e^{-1} H_{\text{attn}}^{k^T} D_v^{-\frac{1}{2}} X P^k) \end{aligned} \tag{12}$$

The hypergraph attention mechanism in Equation (12) captures the importance of a hyperedge q to a node $p \in q$. It calculates the attention coefficient $H_{\text{attn}}(p, q)$, as indicated in Equation (13), where x_p and x_q denote the node and hyperedge features, respectively. The hyperedge feature x_q is generated by summing the feature vectors of the nodes that are contained in the hyperedge q. In Equation (13), a and W are learnable parameters, and $N(p)$ is the set of hyperedges to which p belongs.

$$H_{\text{attn}}(p, q) = \frac{\exp(\text{LeakyReLU}(a\left[W x_p \| W x_q\right]))}{\sum_{k \in N(p)} \exp(\text{LeakyReLU}(a\left[W x_p \| W x_k\right]))} \tag{13}$$

H. CLASSIFICATION AND REGRESSION MODELS

The classification and regression models, the architectures of which are illustrated in Fig. 3, are used to calculate the investment ratio. The classification and regression models use the output of the temporal modeling as the input.

For each stock, the classification model predicts the relative movement between the stock and market index, such as the S&P 500 index. The one-day return rate r_t^i of stock i is compared with the one-day return rate r_t^m of market index m at day t, as indicated in Equation (14). We formulate the movement prediction as a binary classification task, where y_t^i in Equation (14) denotes the ground-truth label at day t. We predict the probability $\hat{y}_t^i$ using a final linear layer with a sigmoid function. The loss function is the binary cross-entropy between the predicted probability and ground-truth label. The predicted probability is used in line 4

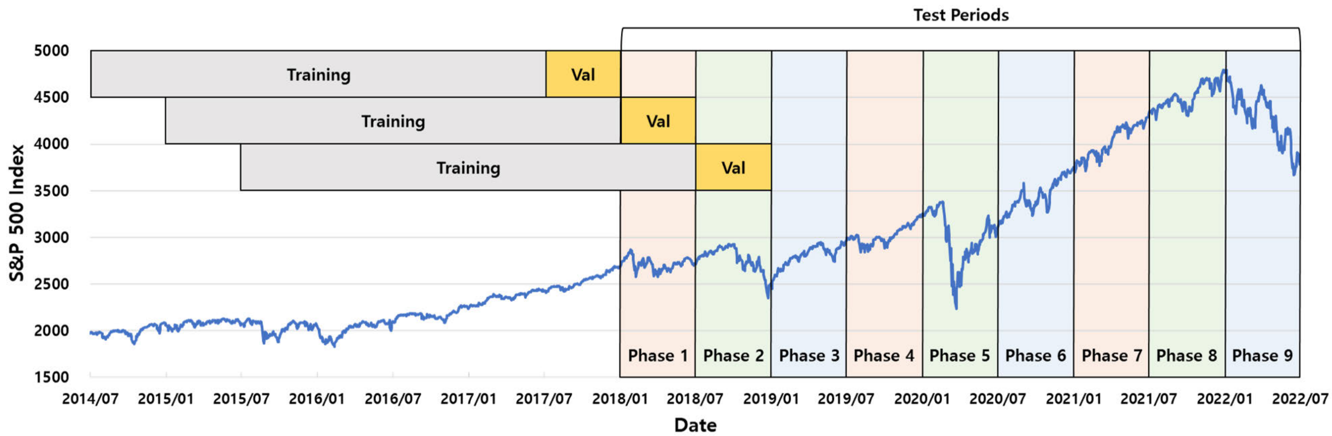

FIGURE 9. Experimental dataset.

of Algorithm 1.

$$y_t^i = \begin{cases} 0 & \text{if } r_t^i \le r_t^m \\ 1 & \text{if } r_t^i > r_t^m \end{cases} \tag{14}$$

The regression model predicts the closing price of the stocks on the next day using a final linear layer. The loss function is the mean squared error between the predicted and real prices. The predicted price is used to calculate the predicted return rate in line 5 of Algorithm 1.

I. DISCUSSION

Table 4 compares the proposed portfolio management framework, namely ASA, with deep learning-based state-of-the-art methods in three dimensions: temporal modeling, stock selection, and stock allocation. RSR [13] does not use Hawkes attention for temporal modeling or hypergraphs for stock selection, and it does not consider stock allocation. STH [14] does not use simple graphs for stock selection and it does not consider stock allocation. HATS [6] does not use Hawkes attention for temporal modeling or hypergraphs and ranking for stock selection, and it does not consider stock allocation. AFHGN [19] does not use Hawkes attention for temporal modeling or simple graphs for stock selection, and it does not consider stock allocation. Thus, to the best of our knowledge, ASA is the only deep learning-based method that combines stock selection with stock allocation, hybridizes simple graph- and hypergraph-based ranking models, and integrates all techniques (LSTM, Hawkes attention, simple graph, hypergraph, ranking, classification, and regression) corresponding to the three dimensions.

TABLE 4. Comparison of portfolio management methods.

	Temporal modeling		Stock selection			Stock allocation	
	RNN	Hawkes attention	simple graph	hyper-graph	ranking	classifi-cation	regres-sion
ASA	LSTM	○	○	○	○	○	○
RSR [13]	LSTM	×	○	×	○	×	×
STH [14]	LSTM	○	×	○	○	×	×
HATS [6]	LSTM	×	○	×	×	×	×
AFHGN [19]	GRU	×	×	○	○	×	×

V. EXPERIMENTS

A. EXPERIMENTAL SETUP

1) DATASETS

The performance of the proposed portfolio management framework was evaluated using stocks that were included in the S&P500 index. Stock data consisting of the opening, high, low, closing, and volume values, as well as economic data, such as commodities, bonds, currencies, and market indices, were collected from Yahoo Finance [46]. Although the stock market tends to rise in the long term, the volatility of the market is very high in certain periods (e.g., phase 5 in Fig. 9). As in [6], we divided the test dataset into smaller test datasets corresponding to different phases to consider diverse market conditions, as illustrated in Fig. 9. The average training, validation, and test periods per phase were 756, 126, and 126 days, respectively. The validation data were used to select the model with minimal validation loss. We generated simple graphs and hypergraphs on the start day of the validation period for each phase based on stocks that remained in the S&P500 index for the validation and test periods. All experiments were repeated 10 times for each phase, and the median results were reported.

To simulate real and practical trading, we used only the long position, started with an initial balance of \$10,000, and considered a transaction cost rate of 0.1%, unless otherwise specified. For simplicity, we bought the maximum number of shares of stocks following the investment ratios and sold all shares of stocks at the closing price on the next day.

2) EVALUATION METRICS

We evaluated the performance of each method in terms of the return rate and risk indicators (i.e., the Sharpe ratio (SR) [47] and maximum drawdown (MDD) [34]). The return rate was calculated using $(PV_{end} - PV_{start})/PV_{start}$, where PV_{start} and

PV_{end} are the initial balance and the portfolio value after the test period, respectively.

The SR measures the return of an investment compared to its risk, as expressed by Equation (15), where $\mathbb{E}[R]$ is the expected return and $\sigma[R]$ is the standard deviation of the return, which measures the fluctuations (i.e., risk). Higher values of SR correspond to higher risk-adjusted returns. In Equation (15), the change rate of the portfolio value is used as the return.

$$\mathrm{SR} = \frac{\mathbb{E}[R]}{\sigma[R]} \tag{15}$$

The MDD measures the maximum loss rate from the peak to the trough of a portfolio over a specified period T, as indicated in Equation (16). The drawdown for time τ is computed using the inner maximum term. Lower MDD scores correspond to lower risk.

$$\mathrm{MDD}(T) = \max_{\tau \in (0,T)} \left[\max_{t \in (0,\tau)} \frac{PV_t - PV_\tau}{PV_t} \right] \tag{16}$$

3) BASELINE METHODS

The deep learning-based state-of-the-art methods listed below, for which the source code has been released, were compared with the proposed ASA method to ensure a fair evaluation. We measured the performance for the top-5, 10, and 15 stocks and reported the best performance for all the methods, as portfolio management studies have demonstrated that approximately 10 stocks are needed to reduce risks [48]. The simple graphs and hypergraphs that were constructed in this study exhibited the best performance for all methods. We used the Adam optimizer with momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, epsilon $= 10^{-7}$, decay $= 0.99$, learning rate $= 0.001$, and mini-batch size $= 64$. The network structures and hyperparameters were optimized for each method, as described below, and the others were identical to those in the original paper.

- The **buy and hold (B&H)** method buys the S&P500 index on the first day of each test phase, holds it, and sells it on the last day of the phase.
- **HATS [6]** uses LSTM and the hierarchical attention mechanism. We set the number of LSTM units to 64, the dropout rate to 0.5, and the number of top-K stocks to 5.
- **STH [14]** uses the Hawkes attention mechanism and hypergraph convolution with a multi-head attention mechanism, which are described in Sections IV-E and IV-G, respectively. We set the sliding window size to 16 and the number of top-K stocks to 5.
- **RSR [13]** uses LSTM and temporal graph convolution. We set the number of LSTM units to 64, the sliding window size to 16, and the number of top-K stocks to 5.
- **RSR+S** is an enhanced method based on RSR to verify the effectiveness of the simple stock allocation method that linearly increases the investment ratio based on the ranking score of the stocks that are selected via RSR.

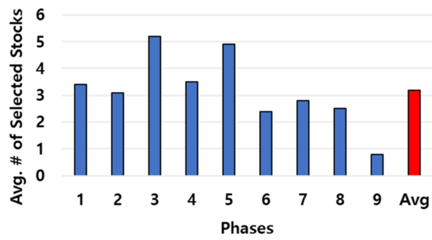

FIGURE 10. Average number of selected stocks of ASA for each phase.

- **RSR+C&R** is an enhanced RSR method that applies the proposed stock allocation method to determine the investment ratio, as in Algorithm 1, using the classification and regression models.

4) IMPLEMENTATION DETAILS OF ASA

In the data preprocessing, we set the distance threshold for the hierarchical clustering to 0.5, the feature selection threshold F to 8, the estimator for the feature selection to linear regression, the reduced dimensionality to 16, and K in the top-K for the node feature selection to 12. In the temporal modeling, we set the sliding window sizes for the stock selection and allocation to 20 and 96, respectively, the number of LSTM units for the ranking models to 64, and the number of Bi-LSTM units for the classification and regression models to 128. In the simple graph-based ranking model, we set the number K of heads in Equation (5) to 4, ρ in Equation (11) to 0.1, and K in the top-K selected stocks to 10. In the hypergraph-based ranking model, we set the number of channels for the HConv layers to 32, the number K of heads for the first and second HConv layers in Equation (12) to 4 and 1, respectively, ρ in Equation (11) to 0.1, and K in the top-K selected stocks to 10.

B. EXPERIMENTAL RESULTS

1) COMPARISON WITH OTHER METHODS

As indicated in Table 5, ASA outperformed all other methods in all phases except for phase 5 in terms of the return rate and SR. The improvements in ASA were statistically significant in a t-test ($p < 0.01$). ASA achieved an average return rate of 27.0%, which was 14.6%P higher than that of the second-best method, namely RSR. In particular, for phase 9 trending downwards, all methods suffered significant losses except for ASA, which recorded a good profit. This is because ASA can select a variable number of stocks depending on the market conditions by intersecting the top-K stocks that are obtained from the two ranking models SG and HG in Algorithm 1, whereas the other methods select a fixed number of stocks using only a single model. As illustrated in Fig. 10, ASA selected less than one stock per day on average in phase 9 because few profitable stocks existed when the market was trending downwards.

In terms of risk indicators, the average SR of ASA was 1.29, which was 0.73 higher than that of the second-best

TABLE 5. Experimental results for each phase.

Phases	Return rate					Sharpe ratio (SR)					Maximum drawdown (MDD)				
	B&H	HATS	STH	RSR	ASA	B&H	HATS	STH	RSR	ASA	B&H	HATS	STH	RSR	ASA
1	0.9%	2.4%	8.7%	-2.3%	**17.7%**	-0.13	0.24	0.66	-0.25	**1.12**	0.10	0.10	0.09	**0.09**	0.12
2	-8.9%	-9.2%	-10.9%	-7.4%	**11.6%**	-0.93	-0.99	-0.75	-0.36	**0.82**	0.20	0.22	0.21	0.21	**0.07**
3	17.1%	20.6%	19.7%	10.7%	**31.8%**	1.49	1.85	1.88	0.69	**2.04**	**0.07**	0.10	0.12	0.18	0.08
4	8.3%	9.7%	10.5%	20.9%	**24.5%**	0.60	0.75	0.77	1.26	**1.31**	**0.06**	0.09	0.14	0.09	0.11
5	-5.1%	-8.2%	-6.5%	**69.1%**	12.8%	-0.11	-0.15	-0.14	**1.28**	0.44	**0.34**	0.38	0.37	0.35	0.36
6	19.1%	24.8%	42.8%	26.6%	**64.6%**	1.29	1.44	1.48	1.62	**1.63**	**0.10**	0.11	0.14	0.11	0.16
7	16.0%	21.3%	33.1%	25.6%	**49.0%**	1.31	1.39	1.76	1.60	**2.30**	**0.04**	0.07	0.11	0.10	0.06
8	9.8%	3.4%	14.9%	1.3%	**24.4%**	0.74	0.61	0.77	0.26	**0.79**	**0.05**	0.08	0.13	0.16	0.11
9	-21.0%	-17.5%	-38.2%	-33.3%	**6.2%**	-1.43	-1.08	-2.57	-1.06	**1.16**	0.23	0.22	0.39	0.38	**0.05**
Min.	-21.0%	-17.5%	-38.2%	-33.3%	**6.2%**	-1.43	-1.08	-2.57	-1.06	**0.44**	**0.04**	0.07	0.09	0.09	0.05
Max.	19.1%	24.8%	42.8%	**69.1%**	64.6%	1.49	1.85	1.88	1.62	**2.30**	**0.34**	0.38	0.39	0.38	0.36
Avg.	4.0%	5.3%	8.2%	12.4%	**27.0%**	0.31	0.45	0.43	0.56	**1.29**	0.13	0.15	0.19	0.19	**0.12**
Std.	**0.14**	0.15	0.24	0.28	0.19	1.03	1.05	1.41	0.96	**0.61**	0.10	0.10	0.11	0.11	**0.09**

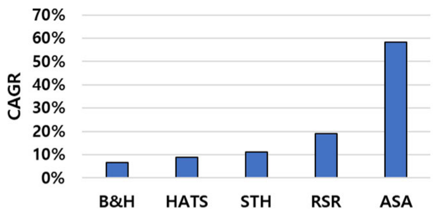

FIGURE 11. Comparison with other methods.

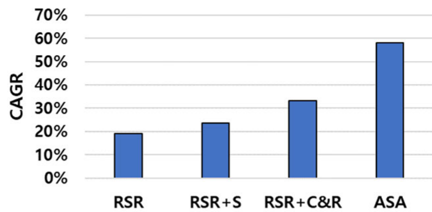

FIGURE 12. Comparison with simple stock allocation method.

method, namely RSR. The average MDD of ASA was 0.12, which was 0.01 lower than that of the second-best method, namely B&H. For phase 5, ASA exhibited the second-best return rate and RSR was the best. This is because RSR focuses on maximizing profits rather than reducing risks, as indicated in the average performance of the risk indicators (SR and MDD). Compared to RSR, ASA exhibited well-balanced profit and risk by combining various different stock selection and allocation models.

Fig. 11 depicts the compounded annual growth rate (CAGR) of the methods over the entire test period. ASA outperformed all other methods in terms of the CAGR, achieving a CAGR of 58.2%, which was 39.1%P higher than that of the second-best method, namely RSR. This is because ASA achieved good profits even when the market was declining in phases 2, 5, and 9, resulting in a compound interest effect. As indicated in Table 6, ASA exhibited a 7%P lower number of loss days compared with the second-best method, which reduced the negative impact of the losses on compounding.

In summary, the results demonstrate that ASA, which combines stock selection with stock allocation, hybridizes simple graph- and hypergraph-based ranking models, and integrates all of the techniques in Table 4, is effective for portfolio management.

2) COMPARISON WITH SIMPLE STOCK ALLOCATION METHOD

Fig. 12 depicts the effects of the stock allocation methods. RSR+S exhibited a higher CAGR than RSR, but did not show a significant improvement (only 4.5%P). RSR+C&R achieved a significant improvement over RSR (14.2%P). These results indicate that stock allocation is effective in portfolio management, but that the simple method, which linearly increases the investment ratio based on the ranking score, is not sufficient for maximizing profits. Furthermore, ASA exhibited a 24.9%P higher performance than RSR+C&R, which demonstrates the effectiveness of our framework.

TABLE 6. Number of days of investment during the test period.

Methods	Profit days	Loss days	No trade	Total
ASA	562 (50%)	463 (41%)	106 (9%)	1131
RSR	588 (52%)	543 (48%)	0 (0%)	1131

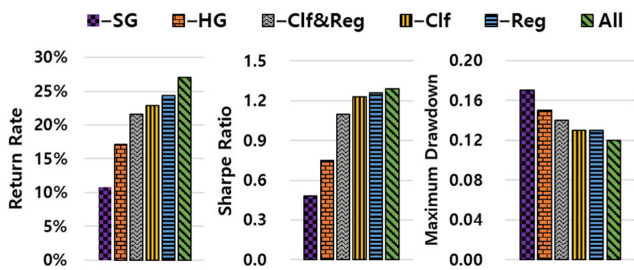

FIGURE 13. Experimental results of the ablation studies.

3) ABLATION STUDIES

We conducted ablation studies to evaluate the contribution of each ASA component. The excluded components were the simple graph (SG), hypergraph (HG), classification

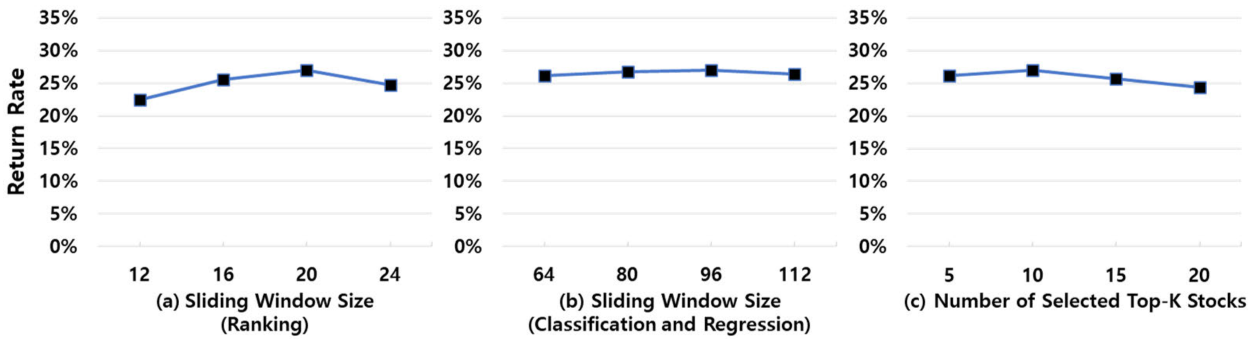

FIGURE 14. Comparison with other hyperparameter values.

TABLE 7. List of the relation types used in this study.

Code	Relation Name	Description
P17	Country	sovereign state of this item; don't use on humans.
P31	Instance of	that class of which this subject is a particular example and member (subject typically an individual member with a proper name label).
P112	Founded by	founder or co-founder of this organization, religion or place.
P113	Airline hub	airport that serves as a hub for an airline.
P114	Airline alliance	alliance the airline belongs to.
P121	Item operated	equipment, installation or service operated by the subject.
P127	Owned by	owner of the subject.
P131	Located in the administrative territorial entity	this item is located on the territory of the fallowing administrative entity.
P155	Follows	immediately prior item in a series of which the subject is a part.
P156	Followed by	the immediately following item in some series of which the subject is part.
P159	Headquarters location	specific location where an organization's headquarters is or has been situated.
P166	Award received	award or recognition received by a person, organisation or creative work.
P169	Chief executive officer	highest-ranking corporate officer appointed as the CEO within an organization.
P176	Manufacturer	manufacturer or producer of this product.
P199	Business division	divisions of this organization.
P306	Operating system	operating system (OS) on which a software works.
P355	Has subsidiary	subsidiary of a company or organization, opposite of parent organization.
P361	Part of	object of which the subject is a part.
P366	Has use	main use of the subject.
P400	Platform	platform for which a work was developed or released, or the specific platform version of a software product.
P414	Stock exchange	exchange on which this company is traded.
P452	Industry	industry of company or organization.
P463	Member of	organization or club to which the subject belongs.
P495	Country of origin	country of origin of this item (creative work, food, phrase, product, etc.).
P625	Coordinate location	geocoordinates of the subject.
P740	Location of formation	location where a group or organization was formed.
P749	Parent organization	parent organization of an organisation, opposite of subsidiaries.
P793	Significant event	significant or notable events associated with the subject.
P1056	Product or material produced	material or product produced by a government agency, business, industry, facility, or process.
P1344	Participant in	event a person or an organization was/is a participant in.
P1454	Legal form	legal form of an organization.
P1830	Owner of	entities owned by the subject.
P1889	Different from	item that is different from another item, with which it is often confused.
P2770	Source of income	source of income of an organization or person.
P3320	Board member	member(s) of the board for the organization.
P4950	Irish rugby football union women's sevens player ID	identifier for a women's rugby sevens player selected with the Ireland national team on the Irish Rugby Football Union website.
P5009	Complies with	the product or work complies with a certain norm or passes a test.
P6379	Has works in the collection	collection that have works of this artist.

(Clf), regression (Reg), and both classification and regression (Clf&Reg). ASA with all components is denoted by "All." Fig. 13 depicts the average performances of the various models. The results reveal that the aforementioned components contributed to the performance improvement in the following order: SG, HG, Clf&Reg, Clf, and Reg. In particular,

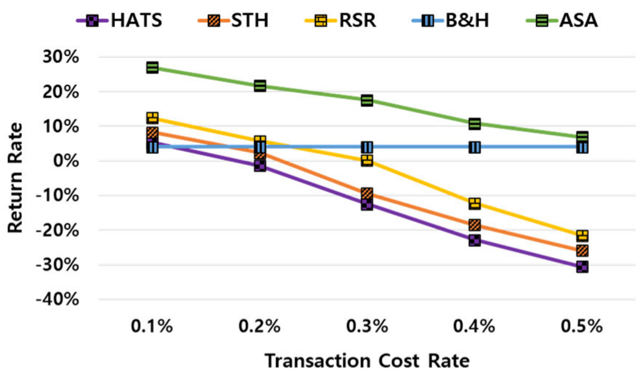

FIGURE 15. Experimental results of the robustness study.

the average return rate was 5.4%P poorer when the stock allocation (i.e., Clf&Reg) was excluded. These results indicate that all components complement one another.

4) COMPARISON WITH OTHER HYPERPARAMETER VALUES

Fig. 14 presents the effects of important hyperparameters on the ASA performance, including the sliding window size for the ranking models, that for the classification and regression models, and K in the top-K stocks that were selected from each ranking model. As illustrated in Fig. 14(a), the ASA performance was degraded when the sliding window size for the ranking models was too small or too large. This is because the information is insufficient (excessive) when it is too small (large). Fig. 14(b) indicates that ASA was not sensitive to the sliding window size for the classification and regression models. Fig. 14(c) reveals that the ASA performance was degraded as K increased. This is because low-profit stocks were included in the portfolio when K was large.

5) ROBUSTNESS STUDY

Transaction costs (e.g., transaction fees and taxes) exist in a real trading environment. We evaluated the effect of the transaction cost rate on the ASA performance to verify its robustness. Fig. 15 depicts the average return rate with respect to varying transaction cost rates. The results reveal that the average return rate decreased as the transaction cost rate increased for all methods. However, ASA achieved the best performance even when the transaction cost rate was 0.5%, which is higher than that of real trading environments. This is because the profit per trade of ASA is higher than the transaction cost per trade.

VI. CONCLUSION AND FUTURE WORK

We have proposed a novel deep learning-based framework called ASA that integrates stock selection and allocation for effective and autonomous portfolio management. We hybridized the simple graph and hypergraph-based ranking models to select the most profitable stocks. We combined the classification and regression models to determine the investment ratio. Furthermore, we proposed a robust feature extraction method for stock and economic data and incorporated LSTM, Bi-LSTM, and the Hawkes attention mechanism for temporal modeling. We compared the performance of ASA with that of state-of-the-art methods. The results of experiments on stocks that were included in the S&P500 index revealed that ASA achieved a CAGR of 58.2%, which was 39.1%P higher than that of the second-best method. In particular, ASA reduced the losses by selecting a lower number of stocks for a declining market. In future work, we plan to incorporate more stock selection and allocation models as well as various graph data into our framework.

APPENDIX

See Tables 7 and 8.

TABLE 8. List of abbreviations.

Abbreviation	Description
ASA	Autonomous stock Selection and Allocation
Bi-LSTM	Bidirectional Long Short-Term Memory
CAGR	Compounded Annual Growth Rate
GAT	Graph Attention Network
GNN	Graph Neural Network
GRU	Gated Recurrent Unit
HConv	Hypergraph Convolution
HG	Hypergraph
LSTM	Long Short-Term Memory
SG	Simple Graph

REFERENCES

[1] G. R. Babu, *Portfolio Management: Including Security Analysis*. Mohan Garden, India: Concept Publishing Company, 2007.
[2] C.-H. Kuo, C.-T. Chen, S.-J. Lin, and S.-H. Huang, "Improving generalization in reinforcement learning–based trading by using a generative adversarial market model," *IEEE Access*, vol. 9, pp. 50738–50754, 2021.
[3] R. K. Nayak, D. Mishra, and A. K. Rath, "A naïve SVM-KNN based stock market trend reversal analysis for Indian benchmark indices," *Appl. Soft Comput.*, vol. 35, pp. 670–680, Oct. 2015.
[4] L. Khaidem, S. Saha, and S. R. Dey, "Predicting the direction of stock market prices using random forest," 2016, *arXiv:1605.00003*.
[5] L. Lai, C. Li, and W. Long, "A new method for stock price prediction based on MRFs and SSVM," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2017, pp. 818–823.
[6] R. Kim, C. Ho So, M. Jeong, S. Lee, J. Kim, and J. Kang, "HATS: A hierarchical graph attention network for stock movement prediction," 2019, *arXiv:1908.07999*.
[7] R. Sawhney, S. Agarwal, A. Wadhwa, and R. R. Shah, "Spatiotemporal hypergraph convolution network for stock movement forecasting," in *Proc. ICDM*, 2020, pp. 482–491.
[8] X. Yuan, J. Yuan, T. Jiang, and Q. U. Ain, "Integrated long-term stock selection models based on feature selection and machine learning algorithms for China stock market," *IEEE Access*, vol. 8, pp. 22672–22685, 2020.
[9] J. Ye, J. Zhao, K. Ye, and C. Xu, "Multi-graph convolutional network for relationship-driven stock movement prediction," in *Proc. ICPR*, 2021, pp. 6702–6709.
[10] Q. Chen and C.-Y. Robert, "Graph-based learning for stock movement prediction with textual and relational data," *J. Financial Data Sci.*, vol. 4, no. 4, pp. 152–166, Sep. 2022.
[11] J. Sun, "A stock selection method based on earning yield forecast using sequence prediction models," 2019, *arXiv:1905.04842*.

[12] F. Yang, Z. Chen, J. Li, and L. Tang, "A novel hybrid stock selection method with stock prediction," *Appl. Soft Comput.*, vol. 80, pp. 820–831, Jul. 2019.
[13] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T.-S. Chua, "Temporal relational ranking for stock prediction," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, pp. 1–30, Apr. 2019.
[14] R. Sawhney, S. Agarwal, A. Wadhwa, T. Derr, and R. R. Shah, "Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 1, 2021, pp. 497–504.
[15] J. Gao, X. Ying, C. Xu, J. Wang, S. Zhang, and Z. Li, "Graph-based stock recommendation by time-aware relational attention network," *ACM Trans. Knowl. Discovery From Data*, vol. 16, no. 1, pp. 1–21, Feb. 2022.
[16] S. Feng, C. Xu, Y. Zuo, G. Chen, F. Lin, and J. XiaHou, "Relation-aware dynamic attributed graph attention network for stocks recommendation," *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108119.
[17] Y. He, Q. Li, F. Wu, and J. Gao, "Static-dynamic graph neural network for stock recommendation," in *Proc. 34th Int. Conf. Sci. Stat. Database Manage.*, Jul. 2022, pp. 1–4.
[18] H. Wang, T. Wang, S. Li, J. Zheng, S. Guan, and W. Chen, "Adaptive long-short pattern transformer for stock investment selection," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 3970–3977.
[19] X. Ma, T. Zhao, Q. Guo, X. Li, and C. Zhang, "Fuzzy hypergraph network for recommending top-K profitable stocks," *Inf. Sci.*, vol. 613, pp. 239–255, Oct. 2022.
[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[21] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*.
[22] G. Pacreau, E. Lezmi, and J. Xu, "Graph neural networks for asset management," [Online]. Available: https://ssrn.com/abstract=3976168
[23] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
[24] B. Yan, C. Wang, G. Guo, and Y. Lou, "TinyGNN: Learning efficient graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1848–1856.
[25] S. Jin, C. Jing, Y. Wang, and X. Lv, "Spatiotemporal graph convolutional neural networks for metro flow prediction," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 43, pp. 403–409, 2022.
[26] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[27] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. ICLR*, 2018, pp. 1–12.
[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[29] X. Zhao, Y. Liu, Y. Xu, Y. Yang, X. Luo, and C. Miao, "Heterogeneous star graph attention network for product attributes prediction," *Adv. Eng. Informat.*, vol. 51, Jan. 2022, Art. no. 101447.
[30] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.
[31] Y. Ye, H. Pei, B. Wang, P.-Y. Chen, Y. Zhu, J. Xiao, and B. Li, "Reinforcement-learning based portfolio management with augmented asset movement prediction states," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, 2020, pp. 1112–1119.
[32] P. Koratamaddi, K. Wadhwani, M. Gupta, and S. G. Sanjeevi, "Market sentiment-aware deep reinforcement learning approach for stock portfolio allocation," *Eng. Sci. Technol., Int. J.*, vol. 24, no. 4, pp. 848–859, Aug. 2021.
[33] D.-Y. Park and K.-H. Lee, "Practical algorithmic trading using state representation learning and imitative reinforcement learning," *IEEE Access*, vol. 9, pp. 152310–152321, 2021.
[34] S.-H. Kim, D.-Y. Park, and K.-H. Lee, "Hybrid deep reinforcement learning for pairs trading," *Appl. Sci.*, vol. 12, no. 3, p. 944, Jan. 2022.
[35] Q. Y. E. Lim, Q. Cao, and C. Quek, "Dynamic portfolio rebalancing through reinforcement learning," *Neural Comput. Appl.*, vol. 34, no. 9, pp. 7125–7139, May 2022.
[36] P. Ghosh, A. Neufeld, and J. K. Sahoo, "Forecasting directional movements of stock prices for intraday trading using LSTM and random forests," *Finance Res. Lett.*, vol. 46, May 2022, Art. no. 102280.
[37] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognit. Lett.*, vol. 15, no. 11, pp. 1119–1125, 1994.
[38] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graph. Statist.*, vol. 15, no. 2, pp. 265–286, Jan. 2006.
[39] D. Fengqian and L. Chao, "An adaptive financial trading system using deep reinforcement learning with candlestick decomposing features," *IEEE Access*, vol. 8, pp. 63666–63678, 2020.
[40] *TA-Lib: Technical Analysis Library*. Accessed: Sep. 11, 2022. [Online]. Available: http://ta-lib.org/
[41] X. Ding, J. Shi, J. Duan, B. Qin, and T. Liu, "Quantifying the effects of long-term news on stock markets on the basis of the multikernel Hawkes process," *Sci. China Inf. Sci.*, vol. 64, no. 9, pp. 1–13, Sep. 2021.
[42] B. Herrendorf, R. Rogerson, and Á. Valentinyi, "Two perspectives on preferences and structural transformation," *Amer. Econ. Rev.*, vol. 103, no. 7, pp. 2752–2789, Dec. 2013.
[43] *Global Industry Classification Standard*. Accessed: Sep. 11, 2022. [Online]. Available: https://www.msci.com/our-solutions/indexes/gics
[44] D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledge-base," *Commun. ACM*, vol. 57, no. 10, pp. 78–85, 2014.
[45] S. Bai, F. Zhang, and P. H. S. Torr, "Hypergraph convolution and hypergraph attention," *Pattern Recognit.*, vol. 110, Feb. 2021, Art. no. 107637.
[46] *Yahoo Finance*. Accessed: Sep. 11, 2022. [Online]. Available: https://finance.yahoo.com/
[47] W. F. Sharpe, "The Sharpe ratio," *J. Portfolio Manage.*, vol. 21, no. 1, pp. 49–58, 1994.
[48] A. Zaimovic, A. Omanovic, and A. Arnaut-Berilo, "How many stocks are sufficient for equity portfolio diversification? A review of the literature," *J. Risk Financial Manage.*, vol. 14, no. 11, p. 551, Nov. 2021.

JAE-SEUNG KIM is currently pursuing the Bachelor of Science degree with Kwangwoon University, Seoul, Republic of Korea.

SANG-HO KIM is currently pursuing the Bachelor of Science degree with Kwangwoon University, Seoul, Republic of Korea.

KI-HOON LEE received the B.S., M.S., and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2000, 2002, and 2009, respectively.

From 2010 to 2012, he was a Manager with the Advanced Institute of Technology, Korea Telecom (KT). From 2012 to 2013, he was a Senior Developer with SAP Labs Korea. He joined Kwangwoon University, in 2013, where he is currently a Professor with the School of Computer and Information Engineering. He has published papers in leading international journals and conferences, including IEEE Access, *The VLDB Journal*, SIGMOD Record, and the IEEE ICDE.